# Near Duplicate Image Identification with Spatially Aligned Pyramid Matching

Dong Xu[1] Tat-Jen Cham[1] Shuicheng Yan[2] Shih-Fu Chang[3]

[1]School of Computer Engineering, Nanyang Technological University, Singapore

[2]Department of Electrical and Computer Engineering, National University of Singapore, Singapore

[3]Department of Electrical Engineering, Columbia University, USA

dongxu@ntu.edu.sg astjcham@ntu.edu.sg eleyans@nus.edu.sg sfchang@ee.columbia.edu

## Abstract

*A new framework, termed Spatially Aligned Pyramid Matching, is proposed for Near Duplicate Image Identification. The proposed method robustly handles spatial shifts as well as scale changes. Images are divided into both overlapped and non-overlapped blocks over multiple levels. In the first matching stage, pairwise distances between blocks from the examined image pair are computed using SIFT features and Earth Mover's Distance (EMD). In the second stage, multiple alignment hypotheses that consider piecewise spatial shifts and scale variation are postulated and resolved using integer-flow EMD. Two application scenarios are addressed – retrieval ranking and binary classification. For retrieval ranking, a pyramid-based scheme is constructed to fuse matching results from different partition levels. For binary classification, a novel Generalized Neighborhood Component Analysis method is formulated that can be effectively used in tandem with SVMs to select the most critical matching components. The proposed methods are shown to clearly outperform existing methods through extensive testing on the Columbia Near Duplicate Image Database and another new dataset.*

## 1. Introduction

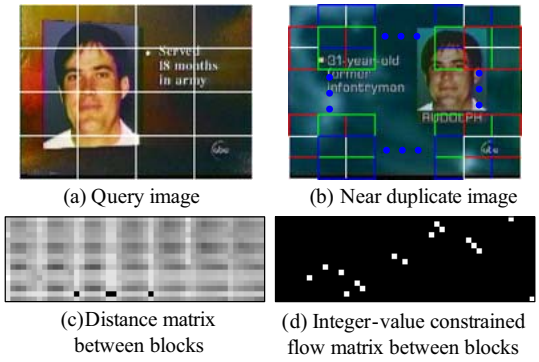
Near duplicate images refer to a pair of images in which one is close to the exact duplicate of the other, but different in conditions related to capture, edits, and rendering. It is a challenging task to identify near duplicate images due to the presence of significant piecewise spatial shifts, scale and photometric variations (See Fig. 1 and 2).

There are two related tasks in Near Duplicate Identification (NDI): Near Duplicate Retrieval (NDR) and Near Duplicate Detection (NDD) [17, 19]. NDR aims to find all images that are near duplicates to an input query image, which can be formulated as a *ranking* problem. NDD aims to detect all duplicate image pairs from all possible pairs from the image source, which can be considered as a two-class *classification* problem. NDR has broad applications in copyright infringement detection and query-by-example application, and NDD has been used to link news stories and group them into threads [17] as well as filter out the redundant near duplicate images in the top results from text keywords based web search [15]. As shown in [17, 19], NDD is more difficult than NDR.

Zhang and Chang [17] formulated a stochastic Attributed Relational Graph (ARG) matching framework for NDI. However, the graph matching method involves a complex process of stochastic belief propagation and thus identification speed is slow [19]. Based on PCA-SIFT, Ke *et al*. [5] developed a point set matching method, while Zhao *et al*. [19] and Wu *et al*. [15] proposed a one-to-one symmetric matching algorithm. However, because of the large number of interest points in images (possibly exceeding 1000), direct matching based on interest points is extremely time-consuming and inappropriate for online NDI. Chum *et al*. [1] addressed large-scale NDI by utilizing both global features and local SIFT descriptors. However, they used a bag-of-words model [6][11] to deal with SIFT features without considering any spatial information.

Distances between images are crucial in NDI. Recently, multi-level matching methods were proposed for efficient distance computation and demonstrated promising results in different tasks, such as object recognition, scene classification and event recognition in news video [3, 6, 7, 16]. They involved pyramidal binning in different domains and led to improved performances resulting from information fusion at multiple levels. The prior work Spatial Pyramid Matching (SPM) [6] used fixed block-to-block matching for scene classification and assumed that images from the same scene have similar spatial configurations. Recently, a multi-level temporal matching technique, referred to as Temporal Pyramid Matching (TPM) [16] here, was proposed to recognize events in broadcast news videos. In TPM, one video

978-1-4244-2243-2/08/$25.00 ©2008 IEEE




(a) Query image (b) Near duplicate image

(c)Distance matrix between blocks (d) Integer-value constrained flow matrix between blocks

Figure 1. Illustration of Spatially Aligned Pyramid Matching at level-2. (a) and (b): A pair of near duplicate images, which are divided into $4 \times 4$ non-overlapped blocks and $7 \times 7$ overlapped blocks (as shown with different colors) respectively. (c): A $16 \times 49$ distance matrix between any two blocks. (d): A $16 \times 49$ integer flow matrix. For better viewing, please see the color pdf file.

clip is divided into non-overlapped subclips, and the subclips across different temporal locations may be matched. However, even when TPM is converted to the spatial domain, TPM cannot cope with the full range of spatial shifts because of its strict non-overlapped partitioning scheme. Moreover, TPM does not consider scale variations.

To solve these problems, in section 2 we propose a two-stage Spatially Aligned Pyramid Matching (SAPM) framework. Images are first divided into multiple tiers of overlapped and non-overlapped blocks. Matching is carried out in two stages, first in which pairwise EMD-based distances are computed between every pair of block signatures comprising SIFT descriptor clusters [18], followed by a second block-alignment stage where different block correspondences at the same level as well as across different levels are hypothesized. The output of SAPM is a set of 45 characteristic multi-level distances, each of which approximately measures the validity of a specific hypothesis, involving spatial shift and scale change. These distances can be used in a single ranking measure for NDR.

In section 3, SAPM is applied to NDD. The multi-level distances are combined as a 45D feature vector for NDD classification. The Generalized Neighborhood Component Analysis (GNCA) method is developed for selecting the most critical matching components prior to SVM learning and classification. Section 4 includes extensive experiments to evaluate SAPM and GNCA. The results show clear superiority of these methods as compared to prior work.

## 2. Spatially Aligned Pyramid Matching

We developed a two-stage matching framework for near duplicate identification. Adopting the approach in SPM [6], we divide an image $x$ into $4^l$ *non-overlapped* blocks at level-$l$, $l = 0, \ldots, L$-1, in which the block size is set as $1/2^l$ of the original image $x$ in both width and height. In this work, we set $L = 3$ based on the empirical observation in [6] that the performance does not increase beyond three levels. Moreover, we consider a finer partition in which overlapped blocks with size equaling $1/2^l$ of the original image (in width and height) are sampled at a fixed interval, say $1/8$ of the image width and height. The denser tiling is intended for subimage matching at finer spatial displacements than that of the non-overlapped partition described above. Fig. 1(a) and (b) illustrate two kinds of partitions. There are a total of five block partition categories, for which we use $p = \{0, 1, 2, 3, 4\}$ to indicate partitions designated as level-0 non-overlapped (L0-N), level-1 non-overlapped (L1-N), level-1 overlapped (L1-O), level-2 non-overlapped (L2-N), and level-2 overlapped (L2-O). The total number of blocks in these five categories are $1, 4, 25, 16$ and $49$ respectively. We represent image $x$ in the $p$-th partition category as $\{x_r^p, r = 1, \ldots, R^p\}$, where $x_r^p$ denotes the $r$-th block and $R^p$ is the total number of blocks. Image $y$ in the $q$-th partition category is represented as $\{y_c^q, c = 1, \ldots, C^q\}$, where $y_c^q$ and $C^q$ are similarly defined. For simplicity, we omit the superscript $p$ and $q$ unless needed.

### 2.1. First Stage Matching

The goal of the first matching stage is to compute the pairwise distances between any two blocks $x_r$ and $y_c$. We represent each block as a bag of orderless SIFT descriptors and specify a distance measure between two sets of descriptors of unequal cardinality. EMD [10] is chosen because of its effectiveness in several different applications [10, 18].

EMD is used to measure the similarity between two signatures $B_1$ and $B_2$. Following [18], we cluster the set of descriptors in block $x_r$ to form its signature $B_1 = \{(\mu_1, w_{\mu_1}), \ldots, (\mu_m, w_{\mu_m})\}$, where $m$ is the total number of clusters, $\mu_i$ is the center of the $i$-th cluster and $w_{\mu_i}$ is the relative size of the $i$-th cluster. Experiments in [18] demonstrated that EMD is relatively robust to the number of clusters in object recognition. In this work, we set $m$ as 40, 20 and 20 respectively for the three different levels. The weight $w_{\mu_i}$ is equivalent to the total supply of suppliers or the total demand of consumers in the original EMD formulation. We also cluster the set of descriptors in block $y_c$ to form its signature $B_2 = \{(\nu_1, w_{\nu_1}), \ldots, (\nu_n, w_{\nu_n})\}$, where $n$ is the total number of clusters, and $\nu_i$ and $w_{\nu_i}$ are defined similarly. We define $d_{ij}$ as the ground distance between $\mu_i$ and $\nu_j$ and use the Euclidean distance as the ground distance in this work because of its simplicity and success in [18]. The EMD between $x_r$ and $y_c$ can be computed by

$$D_{rc} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \widehat{f}_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} \widehat{f}_{ij}} \quad (1)$$

where $\widehat{f}_{ij}$ is the optimal flow that is determined by solving

(a) Query image and blocks (b) Near duplicate image and matched blocks from SPM (c) Near duplicate image and matched blocks from TPM (d) Near duplicate image and matched blocks from our SAPM

Figure 2. Comparison of three pyramid matching methods at level-2. Three blocks in the query images (*i.e.*,(a)) and their matched counterparts in near duplicate images (*i.e.*,(b), (c), (d)) are highlighted and associated by the same color outlines. For better viewing, please see the color pdf file.

the following linear programming problem:

$$\widehat{f}_{ij} = \arg\min_{f_{ij}} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min(\sum_{i=1}^{m} w_{\mu_i}, \sum_{j=1}^{n} w_{\nu_j}); \quad f_{ij} \geq 0;$$

$$\sum_{j=1}^{n} f_{ij} \leq w_{\mu_i}, 1 \leq i \leq m; \sum_{i=1}^{m} f_{ij} \leq w_{\nu_j}, 1 \leq j \leq n. \quad (2)$$

Euclidean distance is a metric and the total weight of each block is constrained to be 1, therefore the EMD distance defined above is a metric [10] (*i.e.*, non-negativity, symmetry and triangle inequality properties hold). The complexity of EMD is $O(m^3 log(m))$ [10] when the total number of clusters in two blocks are the same, *i.e.*, $m = n$. The first matching stage produces the distances between all pairs of blocks. Fig. 1 (c) presents a visual representation of the $16 \times 49$ distance matrix, where brighter intensities indicate higher distance values between corresponding blocks.

## 2.2. Second Stage Matching

The second matching stage aims to align the blocks from one query image $x$ to corresponding blocks in its near duplicate image $y$. Unlike fixed block-to-block matching used in SPM [6], one block may be matched to another block at a different position and/or scale level to robustly handle piecewise spatial translations and scale variations.

Suppose the total number of blocks in $x$ and $y$ are $R$ and $C$, and the pair-wise distances between any two blocks are $D_{rc}$, $r = 1, \ldots, R$ and $c = 1, \ldots, C$. The alignment process involves computing a flow matrix $\widehat{F}_{rc}$ comprising binary elements, which represent unique matches between blocks $x_r$ and $y_c$. For cases when $R = C$, this can be formulated as an integer programming problem embedded within a linear programming framework as suggested by [16]. The following theorem is utilized:

**Theorem 1 ([4])** *The Linear Programming problem,*

$$\widehat{F}_{rc} = \arg\min_{F_{rc}} \sum_{r=1}^{C}\sum_{c=1}^{C} F_{rc} D_{rc}, \; s.t.$$

$$0 \leq F_{rc} \leq 1, \; \forall r, c; \sum_{c=1}^{C} F_{rc} = 1, \; \forall r; \sum_{r=1}^{C} F_{rc} = 1, \; \forall c, \quad (3)$$

*will always have an integer optimum solution when solved with the simplex method*

If $R \neq C$, then assuming that $R < C$ without loss of generality, the EMD formulation for block matching has to be broadened to

$$\widehat{F}_{rc} = \arg\min_{F_{rc}} \sum_{r=1}^{R}\sum_{c=1}^{C} F_{rc} D_{rc}, \; s.t.$$

$$0 \leq F_{rc} \leq 1, \; \forall r, c; \sum_{c=1}^{C} F_{rc} = 1, \; \forall r; \sum_{r=1}^{R} F_{rc} \leq 1, \; \forall c. \quad (4)$$

Nevertheless, the formulation in Eq (3) can be re-established from Eq (4) by 1) adding $C - R$ virtual blocks in image $x$, and 2) setting $D_{rc} = 0$, for all $r$ satisfying $R < r \leq C$. Hence for any solution of Eq (3), a flow matrix for Eq (4) can simply be obtained by removing the elements related to the virtual blocks. An integer solution for Eq (3) with virtual blocks can then be obtained via the simplex method as indicated by Theorem 1, from which the integer solution for Eq (4) may be easily extracted. An outcome of this process is illustrated in Fig. 1(d), indicating the matches of the local image areas in two images (e.g., face, text, etc.).

Fig. 2 compares SPM [6] and TPM [16] with SAPM at level-2, in which three blocks from each query image (*i.e.*, Fig. 2(a)) and their matched counterparts in the near duplicate images (*i.e.*, Fig. 2 (b), (c) and (d)) are highlighted and associated by the same color outlines. Obvious spatial

shifts and scale variations (which also cause spatial shifts) are observable between the near duplicate images. SPM [6] use of fixed block-to-block matching does not handle non-proximal spatial changes. We also converted TPM to the spatial domain to obtain the result in Fig. 2(c), which is equivalent to allowing matching between blocks in different spatial positions across the two compared images. However, the TPM results were still poor as the strict non-overlapped block partitioning scheme does not cope with the full range of spatial changes. When compared with SPM and TPM, results from SAPM were much better, which demonstrates its robustness against spatial shifts and scale variations.

It is worthwhile to point out that SAPM utilizes spatial information because spatial proximity is preserved in higher levels (*e.g.*, level-1 and level-2). An advantage is that the interest points in one spatial block are restricted to match to only interest points within another block in SAPM at a certain level, instead of arbitrary interest points within the entire image as is the case in the classical bag-of-words model (*e.g.*, SPM at level-0) [6].

**Discussions**: 1) Suppose we divide $x$ and $y$ into blocks with the $p$-th and $q$-th partition category respectively, and we denote the *distance measure from* $x^p$ *to* $y^q$ as $S(x^p \rightarrow y^q)$, which can be similarly computed with Eq (1). There are in total 25 distances different variations between the two images: a) If the query image was divided into non-overlapped blocks (*e.g.*, L2-N) and the corresponding database images were divided into overlapped blocks (*e.g.* L2-O) at the same level, spatial shifts and some degree of scale change are addressed (*e.g.*, $S(x^{L2-N} \rightarrow y^{L2-O})$); b) a broad range of scale variations is considered by matching the query image and the database images at different levels[1](*e.g.*, $S(x^{L1-N} \rightarrow y^{L2-O})$; c) Ideally, SAPM can deal with any variations from spatial shift and scale variation by using more denser scales and spatial spacings.

2) From Eq (4), we have the following observations: 1) if $p = q$, $S(x^p \rightarrow y^q) = S(y^p \rightarrow x^q)$; 2) if $p \neq q$, $S(x^p \rightarrow y^q)$ may not be equal to $S(y^p \rightarrow x^q)$. This is obvious because $x^p$ includes different blocks from $x^q$, and also $y^p$ and $y^q$. The two distances are different because the block partitioning schemes are different, hence we describe the distance measure as *asymmetric*.

3) For comparison, we also use another possible weighting scheme, in which normalizing weights $1/R$ and $1/C$ were applied to the two signatures to replace the unit weights 1 in Eq (4). We denote the *distance measure* from $x$ to $y$ in this case as $\widetilde{S}(x^p \rightarrow y^q)$, which is again asymmetric. We will compare the two different weighting schemes for Image NDR in Sec. 4.1.

[1]Subimage cropping is also considered in this work (*e.g.*, $S(x^{L0-N} \rightarrow y^{L1-O})$ and $S(x^{L1-O} \rightarrow y^{LO-N})$). It is treated as a special case of scale variation.

### 2.3. Fusion of Information from Different Levels for NDR

As shown in previous pyramid matching work [3, 6, 7, 16], the best results can be achieved when multiple resolutions are combined, even when results using individual resolutions are not accurate. In this work, we directly fuse the distances from different levels for NDR:

$$S^{\text{Fuse}}(x \rightarrow y) = h_0 S(x^0 \rightarrow y^0) + \sum_{l=1}^{L-1} h_l S(x^{2l-1} \rightarrow y^{2l}), \quad (5)$$

where $h_l$ is the weight for level-$l$. Similar to [16], we tried two weighting schemes: 1) equal weights, and 2) unequal weights. Our experiments demonstrate that the results from different weighting schemes are comparable, similar to the findings obtained for TPM [16].

## 3. Generalized Neighborhood Component Analysis for Near Duplicate Detection

As a *ranking* problem, NDR can be directly conducted based on the distance measures from SAPM. NDR is easier than NDD and amenable to the use of asymmetric distance measures. NDD, conversely, is essentially a two-class classification problem, *i.e.*, an image pair is classified as a duplicate or non-duplicate, which in any case requires symmetric measures. For classification, we need a proper representation for the image pair. A simple solution is to use the difference vector of features in the two images, but we have found that such raw differences are insufficient in detecting duplicate images with large potential variations. Instead, we use 45 matching distances as new input features for the NDD task, with the expectation that near-duplicate image pairs will cluster around the origin in this new feature space while dissimilar image pairs will be far from the origin.

Recall that each weighting scheme in the second stage matching outputs 25 distances, forming a combined 50 distances, except that $S(x^p \rightarrow y^p) = \widetilde{S}(x^p \rightarrow y^p)$ for $p = 0, \ldots, 4$, which means there are only 45 unique distances. Considering that the distance measures are asymmetric, we represent the $k$-th pair of images (say images $x$ and $y$) as two samples, denoted as $t_k^1 \in \mathbb{R}^{45}$ and $t_k^2 \in \mathbb{R}^{45}$, where $t_k^1$ is comprised of the 45 distances from $x$ to $y$, and $t_k^2$ is comprised of another 45 distances from $y$ to $x$. The same class label (1 or 0) is assigned for $t_k^1$ and $t_k^2$. Denote $T$ as the total number of image pairs in the training set, then the training samples are then represented as $\{t_1, \ldots, t_{2T}\} = \{t_1^1, t_1^2, \ldots, t_T^1, t_T^2\}$, and their class labels are denoted as $c_i$, $i = 1, \ldots, 2T$. The index set of samples with same class label as $x_i$ (excluding self) is denoted as $\pi_i = \{j | c_i = c_j, i \neq j\}$. In the test stage, the classification scores based on two samples $t_k^1$ and $t_k^2$ are combined to estimate the likely class. While it is possible to use other

approaches (*e.g.*, average or aggregation in a long vector) to handle the asymmetric matching, we decide to use the dual sample approach mentioned above so that patterns associated with individual feature of the asymmetric pair can be preserved and used to detect near duplicates.

In order to select and appropriately emphasize the most discriminative distance-based features, the feature vector comprising the 45 distances is transformed into a lower dimensional feature space via a matrix $A \in \mathbb{R}^{45\times d}$, where $d$ is the dimension after feature selection. We need to decide the transformation matrix $A$. It plays two roles here, that of feature weighting and selection. Inspired by recent work on Neighborhood Component Analysis [2], we develop a new feature extraction algorithm, called Generalized Neighborhood Component Analysis (GNCA), for this purpose.

Let $\rho_{ij}$ be the probability of sample $x_i$ being assigned the class label of $x_j$ (based on a stochastic nearest neighbors framework [2]) in the transformed feature space,

$$\rho_{ij} = \frac{exp(-\|A^T t_i - A^T t_j\|^2)}{\sum_{k\neq i} exp(-\|A^T t_i - A^T t_k\|^2)}, i \neq j. \quad (6)$$

Thus the probability that sample $i$ is correctly classified is

$$\tau_i = \sum_{j\in\pi_i} \rho_{ij},\ i = 1, \ldots, 2T. \quad (7)$$

The overall goal is to maximize the probability of correct classification for all samples, and a reasonable objective function for GNCA is defined as

$$G(A) = \sum_{i=1}^{2T} \tau_i^{\alpha}, \quad (8)$$

where $0 < \alpha < 1$. If $\alpha$ = 1, then GNCA is exactly NCA.

Setting a lower $\alpha < 1$ value effectively lowers the reduced probability target for the individual samples, such that the final correct-classification probabilities among all samples are more even; *i.e.* a set of samples with mid-range probabilities is preferred to a configuration of highly probable samples mixed with improbable samples. For example, in a two-class classification problem with four training samples, the solutions of $(1.0, 1.0, 0.2, 0.2)$ or $(0.6, 0.6, 0.6, 0.6)$ for $\tau_i$'s have the same value for the objective function with $\alpha$ = 1 and are equally desired; however, setting $\alpha < 1$ (*e.g.*, $\alpha$ = 0.5) will lead to a preference for the latter, resulting in all, rather than only half, the training samples being correctly classified. Let $t_{ij} = t_i - t_j$, then the gradient of $G(A)$ with respect to $A$ can be computed as,

$$\frac{\partial G(A)}{\partial A} = \sum_{i=1}^{2T} \alpha\tau_i^{\alpha-1}\frac{\partial \tau_i}{\partial A}$$
$$= 2\alpha A \sum_{i=1}^{2T}(\tau_i^{\alpha} \sum_{k=1}^{2T} \rho_{ik} t_{ik} t_{ik}^T - \tau_i^{\alpha-1} \sum_{j\in\pi_i} \rho_{ij} t_{ij} t_{ij}^T). \quad (9)$$

Gradient descent is used to search for the best matrix $A$. Based on the computed $A$ from GNCA, the $k$-th pair of images $t_k^1$ and $t_k^2$ are converted into the $d$ dimensional feature space as $\bar{t}_k^1 = At_k^1$ and $\bar{t}_k^2 = At_k^2$. Subsequently, SVM [13] is used for classification. In the testing stage, SVM outputs two decision values $\eta_k^1$ and $\eta_k^2$ for the $k$-th pair of images, and then the final decision value is computed as follows:

$$\eta_k = \frac{0.5}{1 + exp(-\eta_k^1)} + \frac{0.5}{1 + exp(-\eta_k^2)}. \quad (10)$$

## 4. Experiments

We conducted extensive experiments to test SAPM and GNCA. The default dataset used is the Columbia Near Duplicate Image Database [17], in which the images are collected from TRECVID 2003 corpus [12]. We also annotated another near duplicate image database, referred to as New Image Dataset, in which the images are chosen from the key-frames of the TRECVID 2005 and 2006 corpus [12]. Moreover New Image Dataset contains both substantial spatial translations and scale variations. In both datasets, there are 150 near duplicate pairs (300 images) and 300 non-duplicate images. When compared with the synthesized data used in [5], image duplicates in our data set are more challenging, as they are collected from real broadcast news (rather than edits of the same image by the authors). We will make our newly annotated database publicly available.

For performance evaluation in image NDR, we used all near duplicate pairs as queries. For each query, other images were ranked based on computed distances. The retrieval performance was evaluated based on probability of successful top-$k$ retrieval [17, 19], *i.e.*, $P(k) = Q_c/Q$, where $Q_c$ is the number of queries that rank their near duplicates within the top-$k$ positions, and $Q$ is the total number of queries.

Considering that NDD is a two-class classification problem, we used Equal Error Rate (EER) for NDD, which measures the accuracy at which the number of false positives and false negatives are equal. In our experiments, we extracted SIFT features via the Laplacian detector [8]. We use the notation "L2-N → L2-O" to indicate a matching, in which the query and database images are divided as L2-N and L2-O respectively. We may also omit L2 and use "N → O" to indicate matching at any level.

### 4.1. Comparison of SAPM under Different Configurations for Image NDR

We compared SAPM for Image NDR under different overlapped and non-overlapped block partition schemes as well as two weighting schemes in the second stage matching (See Sec. 2.2). Tables 1 and 2 show the top-1 retrieval performances from two weighting schemes (unit and normalized weights) on the Columbia database and New Image

| Query Image | Image in Database | | | | |
|---|---|---|---|---|---|
| | L0-N | L1-N | L1-O | L2-N | L2-O |
| L0-N | **73.7/73.7** | 48.0/37.7 | 65.3/51.7 | 25.3/6.3 | 32.7/9.7 |
| L1-N | 39.0/61.3 | 74.7/74.7 | **78.0**/71.7 | 62.0/20.7 | 65.7/25.3 |
| L1-O | 52.7/61.0 | 56.3/62.3 | 76.0/76.0 | 13.0/14.3 | 54.7/23.3 |
| L2-N | 16.7/46.7 | 46.0/63.3 | 65.3/65.7 | 69.7/69.7 | **79.0**/65.7 |
| L2-O | 17.0/49.3 | 40.0/66.7 | 67.0/71.3 | 52.0/64.3 | 71.0/71.0 |

Table 1. Top-1 retrieval performance (%) with different block partition categories on the Columbia database. Each table cell reports performances (with unit weights) / (with normalized weights).

Dataset respectively. In both databases, the best results at level-1 and level-2 (shown in bold) are obtained from "L1-N → L1-O" and "L2-N → L2-O" with *unit weights* respectively, which will be used as the ***default configuration*** in later experiments. We also have the following observations.

With unit weights, there are four options at each level (*i.e.*, "N → N", "O → N", "N → O" and "O → O"): 1)"N → N" restricts shift distances to be integral multiples of block size, and thus cannot handle shifts that is smaller than the block size; 2) "O → N" may have some of the query blocks matched to empty blocks padded in solving the integer-flow EMD problem, thus losing the information contained in those "lost" query blocks; 3) "N → O" is the most natural matching scheme, which is analogous to the block-based motion estimation method used in the MPEG video compression standard [14].In the integer-flow EMD solution, some blocks in the database image may be "lost" if they are matched to padded empty blocks. But losing information in database images is acceptable since our objective is to find duplicates of the query image, not the database image; 4) Conceptually, "O → O" provides the most flexible matching, and its performance indeed is the second highest among the above four options (as shown in Table 1 and 2). However, it is still less effective than "N → O". One possible explanation is that in this method the number of blocks increases (e.g., from 16 to 49 at level 2), resulting in a higher normalization total flow in EMD matching (denominator in Eq (1)). This makes duplicates of partial image matches less detectable (due to normalized lower EMD matched scores).

We also observe that the matching distances within the same level are consistently better than those across different levels, especially for the Columbia Dataset. However, for the New Image Data Set, cross-level distances are significantly better than for the Columbia Dataset. This can be attributed to the Columbia Dataset having much smaller scale variation than the New Image Dataset. In practice, cross-level distances deal with greater scale variations, and within-level distances address a smaller range of scale variations. Ideally, SAPM can deal with any scale variations with denser scales and spatial spacings. Finally, the results in each diagonal cell of Tables 1 and 2 are the same, because the distances computed by the two different weighting schemes are expected to be identical in these cases.

| Query Image | Image in Database | | | | |
|---|---|---|---|---|---|
| | L0-N | L1-N | L1-O | L2-N | L2-O |
| L0-N | **82.0/82.0** | 62.3/36.0 | 77.0/55.3 | 29.3/5.7 | 41.3/7.0 |
| L1-N | 51.0/72.3 | 79.3/79.3 | **87.7**/80.3 | 71.7/17.3 | 79.3/21.3 |
| L1-O | 68.0/69.7 | 65.0/70.0 | 84.3/84.3 | 13.0/14.7 | 70.3/22.3 |
| L2-N | 26.3/48.0 | 53.0/67.3 | 78.0/76.3 | 64.7/64.7 | **82.7**/68.7 |
| L2-O | 28.7/51.7 | 52.0/71.0 | 79.3/80.7 | 46.3/64.3 | 78.3/78.3 |

Table 2. Top-1 retrieval performance (%) with different block partition categories on New Image dataset. Each table cell reports performances (with unit weights) / (with normalized weights).

| | L0-N → L0-N | L1-N → L1-N (or L1-O) | L2-N→L2-N (or L2-O) |
|---|---|---|---|
| Single-level (SPM) | 73.7 | 76.3 | 73.3 |
| Single-level (TPM) | 73.7 | 74.7 | 69.7 |
| Single-level (SAPM) | 73.7 | **78.0** | **79.0** |
| Multi-level (SPM) | | 76.7 / 76.0 | 78.0 / 77.3 / 77.7 |
| Multi-level (TPM) | | 75.0 / 75.3 | 75.7 / 74.7 / 75.3 |
| Multi-level (SAPM) | | 77.7 / **78.0** | 79.3 / 80.0 / **80.7** |

Table 3. Top-1 retrieval performance (%) comparison of SAPM, SPM and TPM from single level and multiple levels on Columbia database. In the last three rows, the first number is from the equal weighting scheme and the last one or two numbers in each cell are from the unequal weighting scheme when fusing multiple levels.

## 4.2. Comparison of SAPM with SPM and TPM for Image NDR

We compared SAPM with SPM and TPM for cases when matching was done at individual levels as well as when fusing multiple levels. We tried two weighting schemes for cases when multiple resolutions are fused: 1) equal weights, $h_0 = h_1 = h_2 = 1$, and 2) unequal weights: $h_0 = 1$ and $h_1 = 2$ for fusing only the first two levels as well as $h_0 = h_1 = 1$, $h_2 = 2$ and $h_0 = 1$, $h_1 = h_2 = 2$ for fusing all three levels.

The results are listed in Tables 3 and 4, in which the default configuration is used for SAPM at level-1 and 2. The following observations can be made: 1) When compared with SPM and TPM, the results from SAPM are better, either at a single level (*i.e.*, level-1 or level-2) or with multi-level fusion. 2) For SAPM, in all cases better performance can be achieved when multiple resolutions are combined, even for resolutions that are independently poor; moreover, there is no single level that is universally optimal in the two databases. Therefore the best solution is to combine the information from multiple levels in a principled way. 3) For SAPM the results from different weighting schemes are generally comparable, similar to findings from [16]. 4) The results from TPM are worse than SPM, possibly because near duplicate images retain somewhat similar spatial layouts, which fits the SPM model. We also observed that the best result from fusing the first two levels is better than that from fusing all three levels for SPM and TPM in New Image Dataset, which is consistent with prior work [6, 16].

On the Columbia database, our best top-1 retrieval result is 80.7%, higher than the recent result of 79.0% in [19]. Furthermore, [19] involves complex matching of individual interest points (possibly exceeding 1000) in two images.

| | L0-N → L0-N | L1-N → L1-N (or L1-O) | L2-N→L2-N (or L2-O) |
|---|---|---|---|
| Single-level (SPM) | 82.0 | 82.0 | 71.0 |
| Single-level (TPM) | 82.0 | 79.3 | 64.7 |
| Single-level (SAPM) | 82.0 | **87.7** | **82.7** |
| Multi-level (SPM) | | 85.3 / 84.3 | 84.7 / 81.3 / 82.3 |
| Multi-level (TPM) | | 84.0 / 82.7 | 83.0 / 79.3 / 80.7 |
| Multi-level (SAPM) | | 87.3 / **88.0** | **88.3** / 88.0 / 88.0 |

Table 4. Top-1 retrieval performance (%) comparison of SAPM, SPM and TPM from single level and multiple levels on New Image Dataset. In the last three rows, the first number is from the equal weighting scheme and the last one or two numbers are from the unequal weighting scheme when fusing multiple levels.

| | Columbia Database | New Image Dataset |
|---|---|---|
| SPM | $84.8 \pm 2.3$ | $90.1 \pm 1.0$ |
| TPM | $85.7 \pm 1.9$ | $90.1 \pm 1.3$ |
| SAPM | $86.3 \pm 2.6$ | $91.7 \pm 1.1$ |
| SAPM+NCA | $88.8 \pm 1.2$ | $92.6 \pm 2.8$ |
| SAPM+GNCA | $91.2 \pm 1.0$ | $94.4 \pm 2.2$ |

Table 5. Equal Error Rate (EER %) Comparison of algorithms for Image NDD on the Columbia database and Near Image Dataset.

### 4.3. GNCA for Image NDD

Finally we compared the feature selection method GNCA with NCA for Image NDD. As baseline algorithms, we used the 3 distances computed at 3 independent levels from SPM, TPM and SAPM (default configuration) as input features, and further applied SVMs for classification. In SAPM+NCA and SAPM+GNCA, we used NCA and GNCA to convert a 45-dimensional feature into 3D space and then applied SVM. We randomly partitioned the data into training and test sets. All experiments were repeated 10 times with different random training and test samples, with means and standard deviations reported in Table 5. In each run, we used 20 positive and 80 negative samples to train the projection matrices in NCA and GNCA, while another 40 positive and 160 negative samples were used for SVM training. For SPM, TPM and SAPM, all training samples (60 positive and 240 negative) were used for SVM training. The total number of positive and negative test samples are 90 and 4840 respectively. As observed from Table 5: 1) SAPM outperforms SPM and TPM for Image NDD; 2) the best results are from SAPM+GNCA, demonstrating GNCA as an effective feature extraction algorithm to choose the most critical matching components from SAPM, leading to a framework robust to spatial shift and scale variation.

## 5. Conclusions

A multi-level spatial matching framework with two stage matching is proposed to deal with spatial shifts and scale variations for image-based near duplicate identification. For the NDD task, the GNCA algorithm was proposed to aid in feature extraction. Extensive experiments on the Columbia near duplicate database and one new dataset clearly demonstrate the strong potential of SAPM and GNCA. In the future, we will study efficient and effective algorithms for video near duplicate identification.

**Acknowledgements** This material is based upon work funded by Microsoft Research Asia and the U.S. Government. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government. Support was also received from the Centre for Multimedia & Network Technology (CeMNet), NTU.

## References

[1] O. Chum J. Philbin, M. Isard and A. Zisserman, *Scalable Near Identical Image and Shot Detection*, CIVR'07. 1

[2] J. Goldberger, S. Roweis, G. Hinton and R. Salakhutdinov, *Neighborhood Component Analysis*, NIPS'04. 5

[3] K. Grauman and T. Darrell, *The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features*, ICCV'05. 1, 4

[4] P. Jensen and J. Bard, *Operations Research Models and Methods*, John Wiley and Sons, 2003. 3

[5] Y. Ke, R. Sukthankar and L. Huston, *Efficient Near Duplicate Detection and Sub-image Retrieval*, ACM Multimedia'04. 1, 5

[6] S. Lazebnik, C. Schmid and J. Ponce, *Beyond Bags of Features, Spatial Pyramid Matching for Recognizing Natural Scene Categories*, CVPR'06. 1, 2, 3, 4, 6

[7] H. Ling and S. Soatto, *Proximity Distribution Kernels for Geometric Context in Category Recognition*, ICCV'07. 1, 4

[8] D. Lowe, *Object Recognition from Local Scale-Invariant Features*, ICCV'99. 5

[9] J. Munkres, *Algorithms for the Assignment and Transportation Problems*, Journal of the Society for Industrial and Applied Mathematics, Vol. 5, No. 1, pp. 32-38, Mar. 1957.

[10] Y. Rubner, C. Tomasi and L. Guibas, *The Earth Mover's Distance as a Metric for Image Retrieval*, International Journal of Computer Vision, Vol. 40, No. 2, pp. 99-121, 2000. 2, 3

[11] J. Sivic and A. Zisserman, *Video Google: A Text Retrieval Approach to Object Matching in Videos*, ICCV'03. 1

[12] TRECVID, *http://www-nlpir.nist.gov/projects/trecvid*. 5

[13] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995. 5

[14] Y. Wang, J. Ostermann and Y.Q. Zhang, *Video Processing and Communications*, Prentice Hall, 2001. 6

[15] X. Wu, A. G. Hauptmann and C.W. Ngo, *Practical Elimination of Near Duplicates from Web Video Search*, ACM Multimedia'07. 1

[16] D. Xu and S.-F. Chang, *Visual Event Recognition in News Video using Kernel Methods with Multi-Level Temporal Alignment*, CVPR'07. 1, 3, 4, 6

[17] D. Zhang and S.-F. Chang, *Detecting Image Near Duplicate by Stochastic Attribute Relational Graph Matching with Learning*, ACM Multimedia'04. 1, 5

[18] J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid, *Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study*, International Journal of Computer Vision, Vol.73, No. 2, pp. 213-238, 2007. 2

[19] W.-L. Zhao, C.-W. Ngo, H.-K. Tan and X. Wu, *Near Duplicate Keyframe Identification with Interest Point Matching and Pattern Learning*, IEEE Transactions on Multimedia, Vol. 9, No. 5, pp. 1037-1048, Aug. 2007. 1, 5, 6